

REMARKS

Applicants respectfully request reconsideration and allowance of all pending claims in view of the above-amendments and the following remarks.

I. CLAIM REJECTIONS UNDER §102(e)

Claims 1-20 were rejected under §102(e) as being allegedly anticipated by Dalton, U.S. Patent No. 6,951,017.

While Applicants continue to believe the previous claims are not anticipated by Dalton, Applicant have amended the independent claims to further clarify the wording of the claims.

A. **Present Application**

In an example of the present application, the intent is to find design files that are located somewhere in a directory structure within a first environment (e.g., a hardware description environment), create a list of those files, and feed them to a tool in a second environment. In one example, the need for hard-coded paths in shell script usage is removed. For example, RTL files could be located in several sub-directories in the directory structure and a designer may want to synthesize them to create a gate-level netlist. The .rmk files are used to define which files are the RTL files. In general, the tool finds these .rmk files, figures out which design files are the RTL files (or whichever design files have been requested to be found/gathered), and returns a list of the RTL files including the full path name to the design file in the existing directory structure within the first environment so that list can be fed to and used by the synthesis tool in the second environment for accessing the design files.

Referring to claim 1, the directory structure is parsed to locate file paths to the description files. Also, the description files are parsed to identify file paths to the design files that are defined by the description file. An index is generated, which correlates each description file and its respective file path in the first environment.

A list of the design files is constructed, which contains design file names and respective full file paths for each of the design files in the first environment. The respective full

file paths are constructed by concatenating the file path of the description file that is identified in an index to the file path of the design file that is defined by the description file.

Thus, in the example discussed above, the synthesis tool (in the second environment) accesses at least one of the design files within the first environment through the respective full file path to the design file in the first environment. This full file path was produced by concatenating the file path of the description file identified in an index to the file path of the design file defined by the description file.

B. Dalton

In Dalton, the intent is to migrate files from one directory structure (design system) to another directory structure (design system). It is like a one-time event.

In Dalton, each input file that contains a reference to a directory path in the previous design system (previous directory structure 100) is modified to reference directory paths in the new or updated design system (new directory structure 102). (Col. 3, lines 57-61 and col. 7, lines 46-50). Once updated, the input file does not contain a reference to the directory path in the previous design system.

Referring to claim 1, Dalton does not disclose parsing a directory structure and a description file to locate design files in a first environment and “constructing a list containing design file names and respective full file paths for each of the design files in the first environment,” as recited in claim 1. Rather, the input file of Dalton is modified to reference directory paths in the new design system.

Thus, Dalton also do not disclose the step of “accessing at least one of the design files within the first environment by a design tool in a second environment,” as recited in claim 1.

Further, Dalton does not disclose constructing such a list wherein the respective full file paths are constructed by concatenating the file path of the description file that is identified in the index to the file path of the design file that is defined by the description file, as recited in claim 1.

Accordingly, Dalton does not disclose “accessing at least one of the design files within the first environment by a design tool in a second environment through the respective full

file path to the design file in the first environment, which was produced by concatenating as defined in claim 1.

With respect to the “concatenating” limitation of claim 9, now incorporated into claim 1, the Office Action refers to col. 4, lines 35-45 (“... generating mapping file ... combine the process”) This paragraph has nothing to do with combining file paths. Rather, it relates to combining separate processes.

The Office Action also refers to col. 4, line 24, where Dalton refers to a modification syntax, “S/OLD/NEW/g”. The “S” means “substitute” and the “g” means “global”. This string is processed to substitute globally each occurrence of “OLD” with “NEW”. It does not reflect a concatenation of any file paths.

Even further, even if Dalton were interpreted as disclosing some sort of concatenating, Dalton does not disclose concatenating the file path of a description file to the file path of a design file that is defined by that description file.

C. Independent Claims 10 and 13

Independent claims 10 and 13 are not anticipated by Dalton for similar reasons as were discussed with respect to claim 1.

Since Dalton does not anticipate each and every element of independent claims 1, 10 and 13, Applicants respectfully request that the rejection of these claims and their respective dependent claims under §102(e) be withdrawn.

The Director is authorized to charge any fee deficiency required by this paper or credit any overpayment to Deposit Account No. 12-2252.

Respectfully submitted,

WESTMAN, CHAMPLIN & KELLY, P.A.

By: /David D. Brush/

David D. Brush, Reg. No. 34,557
900 Second Avenue South, Suite 1400
Minneapolis, Minnesota 55402-3319
Phone: (612) 334-3222 Fax: (612) 334-3312